# **Project "Market Pulse" - Handover & Status Summary**

**Generated on: Saturday, July 5, 2025, at 11:58 AM CEST**

## **1. Project Overview**

**Purpose:** "Market Pulse" is a web application designed to provide hotel performance metrics. The project has two primary goals:

1. **Live Operational Dashboard:** To provide a real-time, "on-the-books" view of a hotel's key performance indicators (KPIs) for the immediate future (e.g., the next 7 days).
2. **Strategic Analysis Tool:** To store historical performance data and enable a "vs. The Market" comparison feature, allowing a hotel to benchmark its performance against the aggregated average of other hotels in the same city.

**Core Technology:** The application is built on a Node.js backend using the Express framework, a vanilla JavaScript frontend, and a Neon PostgreSQL database. It is deployed and hosted on Vercel.

## **2. System Architecture & Workflow**

The project uses a modern, cloud-based architecture that separates local development from the live production environment.

* **Local Machine (/Users/karolmarcu/Documents/market-pulse):** This is the **Test Kitchen**. All new code and features are developed and tested here using nodemon server.js to run a local server that connects to the live Neon database for testing.
* **GitHub (https://github.com/karolrockenue/market-pulse.git):** This is the **Safety Net** and the official "Recipe Book". It is the central source of truth for all code. All changes are saved here using git commit and git push.
* **Vercel (https://market-pulse-five.vercel.app/):** This is the **Live Bakery**. It is connected to the GitHub repository and automatically deploys any new code pushed to the main branch. It hosts the live website and runs the scheduled cron jobs.
* **Neon (ep-wispy-hill-a9gmzm0p-pooler.gwc.azure.neon.tech):** This is the **Pantry**. It is the cloud-hosted PostgreSQL database that stores the historical and forecast data snapshots.

## **3. Key Files & Their Purpose**

* **server.js:** The main backend Express server. It handles API requests from the frontend, communicates with the Cloudbeds API, and queries the Neon database.
* **public/index.html:** The main HTML file for the user interface. It contains the structure for the dashboard, including the two metrics tables.
* **public/script.js:** The frontend JavaScript file. It handles all user interactions, fetches data from the backend API endpoints, and renders the data into the tables on the index.html page.
* **daily-refresh.js:** A specialized backend script designed to be run as a daily cron job on Vercel. Its sole purpose is to fetch the next 365 days of forecast data from the Cloudbeds API and update the Neon database.
* **initial-sync.js:** A special, one-time-use script for onboarding new hotels. It fetches 365 days of historical data and 365 days of future data to fully populate the database for a new client.
* **vercel.json:** The configuration file for Vercel. It defines how to build the application, how to route traffic to the different server files, and the schedule for the cron job.
* **.env:** A local file (not uploaded to GitHub) that stores all the secret credentials (API keys, database URL) for local development. These same key-value pairs are configured in the Vercel project settings for the live environment.

## **4. API & Endpoints**

### **External APIs**

* **Cloudbeds Insights API:** The primary source of hotel metrics. We use Dataset 7: Occupancy to get all relevant performance data.
* **Cloudbeds General API:** Used to fetch static hotel details (/api/v1.1/getHotelDetails).

### **Internal API Endpoints (defined in server.js)**

* **POST /api/explore:** The frontend calls this to get a live 7-day forecast directly from the Cloudbeds API.
* **GET /api/hotel-details:** The frontend calls this on page load to display the property's static information.
* **GET /api/metrics-from-db:** The frontend calls this to get the 7-day forecast from our own Neon database for the comparison table.

## **5. Future Plan**

The immediate next step is to build the "vs. The Market" feature as outlined in the Project Plan\_ Database and \_vs.docx document. This involves:

1. Creating a new page or view for the market comparison dashboard.
2. Seeding the database with mock data for other hotels using CSV files.
3. Developing the backend logic to calculate the market average from the database.
4. Building the frontend UI to display the comparison between "Your Hotel" and "The Market".

## **6. Summary of Today's Session (July 5, 2025)**

Today was a highly productive session focused on moving from a simple local application to a fully deployed, cloud-native solution.

1. **Database Ingestion:** We identified and fixed a critical bug where the initial database script was not correctly aggregating data, leading to incorrect values. We created and tested a robust script (ingest-daily-metrics.js, now split into daily-refresh.js and initial-sync.js) that correctly processes and saves data to the Neon database.
2. **Frontend Comparison:** We successfully built the "sanity check" feature on the index.html page, creating two side-by-side tables: one pulling live data from the API and one pulling from our Neon database.
3. **Debugging:** We systematically debugged several issues, including timezone-related date mismatches and frontend rendering bugs, to ensure the data in both tables was a perfect "like-for-like" match.
4. **Vercel Deployment:** We deployed the entire application to Vercel, including configuring the project, setting up environment variables for the secret credentials, and creating a vercel.json file to handle routing and the daily cron job.
5. **GitHub Safety Net:** We initialized a Git repository, connected it to a new repository on GitHub, and pushed the entire working project, establishing a secure, version-controlled "safety net" for all future development.

## **7. Sensitive Project Data (For Internal Use)**

* **Production URL:** https://market-pulse-five.vercel.app/
* **GitHub Repository:** https://github.com/karolrockenue/market-pulse.git
* **Neon Database URL:** postgresql://neondb\_owner:npg\_4onu9ZrBUydf@ep-wispy-hill-a9gmzm0p-pooler.gwc.azure.neon.tech/neondb?sslmode=require&channel\_binding=require
* **.env file content:**# Cloudbeds API credentials
* CLOUDBEDS\_CLIENT\_ID=rockenue\_be\_cRtJg7K1HSUyBeYkbFLVhDMz
* CLOUDBEDS\_CLIENT\_SECRET=CAdJ8G0eMUPWkXj2t5lSq9ZFKLYfuV4s
* CLOUDBEDS\_REFRESH\_TOKEN=oZz9-j7GvcvAShcM-J3gzm7eOXZnvB62b6yzdPvp36E
* CLOUDBEDS\_PROPERTY\_ID=302817
* # Neon Postgres connection
* DATABASE\_URL=postgresql://neondb\_owner:npg\_4onu9ZrBUydf@ep-wispy-hill-a9gmzm0p-pooler.gwc.azure.neon.tech/neondb?sslmode=require&channel\_binding=require

## **8. Developer's Quick Start & Reminders (For Karol)**

This section covers the common workflow questions for continuing development.

### **How to Get Back to Work (After a Restart)**

If you have switched off your computer and want to start working again, you just need to open the project folder.

1. Open the **Terminal** application.
2. Navigate to the project directory with the command: cd /Users/karolmarcu/Documents/market-pulse
3. Open the folder in your code editor (e.g., Visual Studio Code).

### **How to Start the Local Test Server**

To test your application on your local machine before deploying, you need to run the local server.

1. Make sure you are in the project directory in your terminal.
2. Run the command: nodemon server.js
3. The server will start, and you can view your test site at http://localhost:3000.

### **How to Save Your Work (GitHub Checkpoint)**

Saving a "checkpoint" of your work to your GitHub safety net involves three simple commands. You should do this every time you complete a significant change.

* **Add all changed files:**git add .
* **Create the checkpoint with a descriptive message:**git commit -m "A message describing what you changed"
* **Push the checkpoint to GitHub:**git push

### **The Automatic Cron Job**

Yes, the daily cron job is successfully scheduled.

* **How it's scheduled:** The vercel.json file contains the configuration: "schedule": "0 1 \* \* \*".
* **What it means:** This instructs Vercel to automatically run the daily-refresh.js script every day at **1:00 AM UTC**. This corresponds to 2:00 AM in Poland (CEST) during summer, and 2:00 AM CET during winter. This job is what keeps the forecast data in your Neon database up to date.